

Lập trình Java

BÀI TẬP TUẦN 1: Giới thiệu ngôn ngữ Java

☒ Bài tập cá nhân

☐ Bài tập tự luyện

☐ Bài tập nhóm

☒ Bài tập cần nộp

1. Quy định về việc nộp bài:

1.1 Thời hạn nộp bài: buổi thực hành tiếp theo

1.2 Cách nộp: nộp qua diễn đàn.

1.3 Bài làm được nén lại thành một tập tin (.zip hoặc .rar)

1.4 Nếu làm theo nhóm thì đặt tên: MSSV1-MSSV2-MSSV3-MSSV4.zip (hoặc .rar). Lưu ý MSSV theo thứ tự tăng dần.

1.5 Nếu làm một mình thì MSSV.zip (hoặc .rar)

2. Tham khảo

- Các tham số liên quan tới hàm println

Conversion Character	Type	Example
d	Decimal integer	159
x	Hexadecimal integer	9f
o	Octal integer	237
f	Fixed-point floating-point	15.9
e	Exponential floating-point	1.59e+01
g	General floating-point (the shorter of e and f)	—
a	Hexadecimal floating-point	0x1.fccdp3
s	String	Hello
c	Character	H
b	boolean	true
h	Hash code	42628b2
tx	Date and time	See Table 3–7
%	The percent symbol	%

- Tham khảo phần **Date & Time Format: trang 81, core java 1 (ver 8)**

3. Yêu cầu:

- Viết chương trình xuất ra màn hình câu thông báo “Hello World!!!”
- Nhập vào bán kính của đường tròn, tính chu vi đường tròn đó.

Hướng dẫn: sử dụng hằng số Math.PI, khai báo như sau để nhập 1 số từ bàn phím

```
import java.util.*;
Scanner in = new Scanner(System.in);
int r = in.nextInt();
```

- Tính tổng của dãy số sau: $S(n) = 1^1 + 2^2 + 3^3 + 4^4 + \dots + n^n$

Hướng dẫn: dùng hàm Math.pow

- Nhập vào 2 số a, b. Xuất ra màn hình kết quả a/b (lấy số 3 lẻ)

Hướng dẫn: sử dụng hàm print System.out.printf("%.3f", x);

- Viết chương trình tính tổng của dãy số sau: $S(n) = 1 + 2 + 3 + \dots + n$ (n: nhập từ bàn phím, n nguyên không âm)
- Nhập vào số nguyên dương n, liệt kê tất cả các ước số của n.
- Nhập vào số nguyên dương n, cho biết n có bao nhiêu chữ số?
- Nhập vào số nguyên dương n, kiểm tra số nguyên dương n có phải là số đối xứng hay không?
- Kiểm tra số n có phải là số nguyên tố hay không?

10. Nhập vào chuỗi x, y (Hướng dẫn: sử dụng hàm `in.nextLine()`)
- Cho biết tổng chiều dài của chuỗi x và y. (Hướng dẫn: sử dụng hàm `length`)
 - Cho biết 3 kí tự đầu tiên của chuỗi x (Hướng dẫn: sử dụng hàm `substring`)
 - Cho biết 3 kí tự cuối của chuỗi y
 - Cho biết kí tự thứ 6 của chuỗi x (Hướng dẫn: sử dụng hàm `charAt`)
 - Tạo chuỗi mới gồm 3 kí tự đầu tiên của chuỗi x và 3 kí tự cuối của chuỗi y
 - Kiểm tra 2 chuỗi x, y có bằng nhau hay không (phân biệt chữ hoa, thường)? (Hướng dẫn: sử dụng hàm `equals`)
 - Kiểm tra 2 chuỗi x, y có bằng nhau hay không (không phân biệt chữ hoa, thường)? (Hướng dẫn: sử dụng hàm `equalsIgnoreCase`)
 - Cho biết y có xuất hiện trong x hay không? Nếu có, tại vị trí nào? (Hướng dẫn: sử dụng hàm `indexOf`)
 - Cho biết tất cả các vị trí xuất hiện của y trong x?
11. Nhập vào 2 ngày a, b (kiểu `Calendar`) (Hướng dẫn: sử dụng lệnh `Calendar c = Calendar.getInstance();`)
- So sánh a, b
 - Cho biết ngày tiếp theo của ngày a là ngày mấy?
 - Cho biết ngày trước đó của ngày a là ngày mấy?
 - Cho biết ngày a là ngày thứ bao nhiêu trong năm?
 - Cho biết tháng chứa a (vd: a=8/12/2000, tháng 12) có bao nhiêu ngày?
 - Cho biết năm chứa a có phải là năm nhuận hay không?
12. Tính biểu thức $1234567890 + 9876654321 * 3$ (Hướng dẫn: sử dụng lớp `BigInteger`)
13. Nhập vào 2 mảng số nguyên A (kích thước n)
- Tạo mảng số nguyên B (kích thước m) với các giá trị ngẫu nhiên (Hướng dẫn: sử dụng hàm `Math.random`)
 - Xuất toàn bộ các phần tử của B ra màn hình (Hướng dẫn: sử dụng hàm `Arrays.toString`)
 - Tạo mảng C từ mảng A (hay copy A qua C) (Hướng dẫn: sử dụng hàm `copyOf`)
 - Thay thế phần tử thứ 1->3 của mảng C bằng 3 phần tử cuối của mảng B (Hướng dẫn: sử dụng hàm `System.arraycopy`)
 - Sắp xếp mảng C tăng dần và xuất ra màn hình (Hướng dẫn: sử dụng hàm `Arrays.sort`)
 - Nhập vào số x, kiểm tra x có nằm trong mảng C hay không? (Hướng dẫn: sử dụng hàm `Arrays.binarySearch`)
14. Tạo mảng 2 chiều như sau và xuất ra màn hình
- ```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
```
15. Nhập vào ma trận số nguyên, kích thước m x n
- Tìm giá trị nhỏ nhất của ma trận
  - Tìm giá trị lớn nhất của ma trận
  - Tìm tất cả các số nguyên tố của ma trận
  - Tính tổng tất cả các giá trị của ma trận